Utility Server

User Guide

Contents

1 Introduction

1.1 Document Scope & Assumptions

This document provides an overview and introduction to the use of the Utility Server and a description of the specific computing environment on the Utility Server. The intent of this guide is to provide information that will enable the average user to perform computational tasks on the system. To receive the most benefit from the information provided here, you should be proficient in the following areas:

- Use of the UNIX operating system.
- Use of an editor (e.g., vi or emacs).
- Remote usage of computer systems via network access.
- A selected programming language and its related tools and libraries.

1.2 Obtaining Accounts

Your account on the Utility Server is an unallocated account, and is automatically provided to you when your High Performance Computing Account is requested.

1.3 Requesting Assistance

The Consolidated Customer Assistance Center (CCAC) is available to help users with any problems, questions, or training requirements for our HPC systems. Analysts are on duty Monday - Friday, 8:00 a.m. to 11:00 p.m. Eastern Standard Time.

- Web: http://www.ccac.hpc.mil/
- Email:mailto:Help@ccac.hpc.mil
- Phone: 1-877-CCAC-039 (1-877-222-2039) or 937-255-0679
- Fax: 937-656-9538

1.4 Available Services

Although the Utility Server can serve multiple services, it is expected that users will use the Utility Server for access to the Center-Wide File System, Remote Batch Management Capability (managing jobs NOT running batch jobs), Remote Visualization with access to the General Processing computation Graphics Processing Units, and stability to ensure that this system is available when users need to complete the above processes.

The Utility Server is meant to provide multiple services:

- Hierarchical Storage.
- Center Wide Batch Management Capability.
- Remote Visualization Tools.
- Reliability.

In addition to these services, the Utility Server is capable of providing these functions as well:

- Interactive Usage, including:
  - Debug.
  - Complete pre- and post- processing.
  - Build applications.
- Remote Visualization tasks.
- Batch Usage during low usage times.
- GPGPU.

2 System Configuration

The Utility Server is a mixed node cluster consisting of compute, graphics, large memory, and login nodes. The login and compute nodes are populated with two AMD Opteron 2.3GHz processors. The graphics nodes include an NVDIA Tesla M2050 for graphics acceleration. The large memory nodes feature twice the number of cores and memory as the compute nodes. The Utility Server uses a QDR InfiniBand network as its high-speed interconnect for MPI messages and IO traffic.

2.1 Operating System

The operating system on the Utility Server is Red Hat Enterprise Linux (RHEL) version 5.

2.2 File Systems

This section is intended to provide technical specifications about the file systems accessible from the Utility Server.

2.2.1 /us/*arl*/dsrc/home

This path is also accessible using $HOME. $HOME should be used for day to day items (i.e. small binaries, scripts). $HOME should not be used to run jobs or as an archival location. The italicized portion of the above will be replaced by the

site where you are running (i.e. if you are running at Navy DSRC, $HOME will be /us/navydsrc/home).

2.2.2 /us/*arl*dsrc/work

This path is also set up with the environment variable $WORKDIR. It is suggested that you complete your jobs in this location as there is a larger amount of formatted memory for your use. The italicized portion of the above will be replaced by the site where you are running (i.e. if you are running at Navy DSRC, $WORKDIR will be /us/navydsrc/work).

2.2.3 /cwfs/*arl*dsrc

This path is directed to the Center Wide Filesystem which is meant for short term storage (no longer than 30 days). The environment variable for this is $CENTER. This is accessible from both the Utility Server and the HPC systems. The italicized portion of the above will be replaced by the site where you are running (i.e. if you are running at Navy DSRC, $CENTER will be /cwfs/navydsrc).

2.3 Peak Performance

The Utility Server is rated at 11.51 CPU (without GPGPU) TFLOPS.

2.4 Processors

The Utility Server uses two dual core 2.3 GHz AMD Opteron Magny-Cours processors in its compute nodes.

There are two processors per node, each with eight cores, for a total of 16 cores per node. In addition, these processors have 512KB L2 cache per core and 12MB of L3 cache.

2.5 Memory

The Utility Server uses a shared memory model. Memory is shared among all the cores on a node, but is not shared among the nodes across the cluster.

2.6 Node Information

The Utility Server has three types of nodes available:

- 44 Compute Nodes
  - Two 2.3GHz eight-core AMD Operton (Magny-Cours) CPUs, 128GB Random Access Memory (RAM)
  - 704 total cores

- 22 Graphics Nodes
    - Two 2.3GHz eight-core AMD Operton (Magny-Cours) CPUs, 128GB RAM
    - Nvidia Tesla M2050
    - 352 total cores
- 22 Large Shared Memory Nodes
    - Four 2.3GHz eight-core AMD Opteron (Magny-Cours) CPUs, 256GB RAM
    - 704 total cores

2.6.1 Login Nodes

Login nodes are configured the same as the Compute Nodes with the exception of having 64 GB of RAM.

The login nodes are not to be used for large computational (e.g. memory, IO, long executions) work. All executions that require large amounts of system resources must be sent to the compute nodes, large memory nodes, or graphics nodes by batch job submission.

Node Configuration				
	Login Nodes	**Compute Nodes**	**Graphic Nodes**	**Large Memory Nodes**
Total Nodes	3	44	22	22
Operating System	RHEL v.5	RHEL v.5	RHEL v.5	RHEL v.5
Cores	13 User Accessible Cores	16 Cores	16 Cores	32 Cores
CPU Type	AMD Opteron 6134 Magny-Cours (x2)	AMD Opteron 6134 Magny-	AMD Opteron 6134 Magny-	AMD Opteron 6134 Magny-

		Cours (x2)	Cours GPGPU (x2)	Cours (x44)
Core Speed	2.3 GHz	2.3GHz	2.3GHz	2.3GHz
Memory/Node	64 GB	128GB	128GB	256GB
Interconnect Type	QDR InfiniBand	QDR InfiniBand	QDR InfiniBand	QDR InfiniBand

File Systems		
File System*	**File System Type**	**Formatted Capacity**
/us/*arl*dsrc/work ($WORKDIR)	Panasas PanFS	200 TB
/us/*arl*dsrc/home ($HOME)	Panasas PanFS	20 TB
/cwfs/*arl*dsrc	Panasas PanFS	800 TB

Figure 1: Configuration and File System Types.

* File System information will be for the site you are working at. The italicized item will be replaced with AFRL, ARL, ERDC, NAVY, ARSC, or MHPCC.

3 Accessing the System

3.1 Kerberos

A Kerberos client kit must be installed on your desktop to enable you to get a Kerberos ticket. Kerberos is a network authentication tool that provides secure communication by using secret cryptographic keys. Only users with a valid HPCMP Kerberos authentication can gain access to The Utility Server. More information about installing Kerberos clients on your desktop can be found at the CCAC Support page.

3.2 Logging In

- Kerberized SSH.

```
% ssh us.arl.hpc.mil (for ARL)

% ssh us.afrl.hpc.mil (for AFRL)

% ssh us.arsc.hpc.mil (for ORS)

% ssh us.erdc.hpc.mil (for ERDC)

% ssh us.navy.hpc.mil (for Navy)

% ssh us.mhpcc.hpc.mil (for MHPCC)
```

- Kerberized telnet.

```
% ktelnet us.arl.hpc.mil (for ARL)

% ktelnet us.afrl.hpc.mil (for AFRL)

% ktelnet us.arsc.hpc.mil (for ORS)

% ktelnet us.erdc.hpc.mil (for ERDC)

% ktelnet us.navy.hpc.mil (for Navy)

% ktelnet us.mhpcc.hpc.mil (for MHPCC)
```

- Kerberized rlogin.

```
% krlogin us.arl.hpc.mil (for ARL)

% krlogin us.afrl.hpc.mil (for AFRL)
```

```
% krlogin us.arsc.hpc.mil (for ORS)

% krlogin us.erdc.hpc.mil (for ERDC)

% krlogin us.navy.hpc.mil (for Navy)

% krlogin us.mhpcc.hpc.mil (for MHPCC)
```

3.3 File Transfers

File transfers to DSRC systems must be performed using Kerberized versions of the following tools: scp, ftp, sftp, and mpscp, except file transfers to the local archive system.

Files can also be transferred to the local systems through the CWFS.

4 User Environment

4.1 User Directories

4.1.1 Home Directory

- us/usr/*arl*dsrc/home ($HOME)
- 10 TB quota per user

This mount point is for the home directories and application directories for the Utility Server. The italicized portion should be replaced by the site you are currently working at.

4.1.2 Work Directory

- us/usr/*arl*dsrc/work ($WORKDIR).
- 100 TB quota per user
- Standard HPC Scrubbing in place.

This mount point is the local workspace for the Utility Server. The italicized portion should be replaced by the site you are currently working at.

4.1.3 CWFS Directory

- /cwfs/*arl*dsrc ($CENTER)
- 810 TB quota per user.
- Available on HPC systems and the Utility Server.
- Removal of files after thirty days.

This file system is the center wide file system of the center. Files will have a retention period of 30 days and are removed after that retention period has expired. Users must archive their own data from the CWFS to the archive system. The italicized portion should be replaced by the site you are currently working at.

Files not archived by the user within the CWFS 30 day retention period will be deleted and cannot be retrieved.

4.2 Shells

The following shells are available on the Utility Server: csh, bash, ksh, tcsh, and sh. To request a change of your default shell, contact the Consolidated Customer Assistance Center (CCAC).

4.3 Environment Variables

A number of environment variables are provided by default on all HPCMP high performance computing (HPC) systems. We encourage you to use these variables in your scripts where possible. Doing so will help to simplify your scripts and reduce portability issues if you ever need to run those scripts on other systems. The following environment variables are automatically set in your login environment:

4.3.1 Login Environment Variables

Common Environment Variables	
Option	**Purpose**
$CENTER	The path to the Center Wide Filesystem. Meant for short term storage (no longer than 30 days)
$CSI_HOME	The path to the directory for the following list of heavily used application packages: ABAQUS, Accelrys, ANSYS, CFD++, Cobalt, EnSight, Fluent, GASP, Gaussian, LS-DYNA, MATLAB, and TotalView, formerly known as the Consolidated Software Initiative (CSI) list. Other application software may also be installed here by our staff.
$DAAC_HOME	The path to the directory containing the ezViz visualization software
$HOME	Your home directory on the system
$JAVA_HOME	The path to the directory containing the default installation of JAVA
$PET_HOME	The path to the directory containing the tools installed by the PETTT ACE staff. The supported software

	includes a variety of open source math libraries (see BC policy FY06-01) and open source performance and profiling tools (see BC policy FY07-02).
$SAMPLES_HOME	The path to the Sample Code Repository. This is a collection of sample scripts and codes provided and maintained by our staff to help users learn to write their own scripts. There are a number of ready-to-use scripts for a variety of applications.
$WORKDIR	Your work directory on the local temporary file system (i.e., local high speed disk).

4.3.2 Batch-Only Environment Variables

In addition to the variables listed above, the following variables are automatically set only in your batch environment. That is, your batch scripts will be able to see them when they run. These variables are supplied for your convenience and are intended for use inside your batch scripts.

Batch-Only Environment Variables	
Option	**Purpose**
$BC_CORES_PER_NODE	The number of cores per node for the compute node on which a job is running.
$BC_MPI_TASKS_ALLOC	The number of MPI tasks allocated for a job.
$BC_NODE_ALLOC	The number of nodes allocated for a job.

4.4 Archive Usage

The Utility Server is connected to the Center Wide Filesystem (CWFS) which should be used for short term storage (no longer than 30 days). Data on the Center Wide Filesystem will be removed after 30 days. Users are responsible for archiving data from short term storage (CWFS) to long term storage using SLM.

Data not managed by the user before the end of 30 days will be removed from the CWFS and ***cannot be recovered.***

The Center Wide Filesystem should be accessed using Storage Lifecycle Management (SLM).

Additional information regarding SLM can be found in the following documents:

- Storage Lifecycle Management (SLM) Usage.
- Storage Lifecycle Management (SLM) Quick Reference Guide.

Additional information regarding the commands used in SLM can be found in the above documents, and can also be found using the man pages on the Utility Server at the command prompt (i.e. % man sinit)

5 Program Development

5.1 Available Compilers

The Utility Server has two compiler suites:

- PGI Compiler Suite (default)
- GNU Compiler Suite

5.2 Libraries

Per Baseline Configuration Policy, the following libraries will be available on the Utility Server:

- SCALASCA Scalable trace analysis package
- PDT Source-level auto-instrumentation
- Valgrind Memory management analysis and profiling

5.3 Debuggers

Per the Baseline Configuration Policy, the following debuggers will be available on the Utility Server:

- GNU Project Debugger (GDB)
- Totalview

6 Batch Scheduling

Although the Utility Server has the capability to run batch jobs, ***it is not meant to be used as a method to run batch jobs***. Batch jobs should be reserved for times when Utility Server usage is low (i.e. evenings/weekends), but during high usage times, the Utility Server is reserved for access to the Center-Wide FileSystem, SLM and file management, Remote Visualization, and Job management with Center Wide Job Management.

6.1 Scheduler

The Portable Batch System Professional™ (PBSPro) is currently running on the Utility Server. It schedules jobs and manages resources and job queues, and can

be accessed through the interactive batch environment or by submitting a batch request.

6.2 Queue Information

The following table describes the queues available on the Utility Server:

Queue Name	Max Wall Clock Time	Max Cores Per Job	Comments
Serial	18 Hours		This is the default queue for the Utility Server.
X11			Graphics queue, same priority as Serial and PKI-VNC.
PKIVNC			Graphics queue, same priority as Serial and X11.
Parallel	18 Hours		
Transfer	24 Hours		Meant to transfer data files to prepare job information.

6.3 Interactive Logins

When you log in to The Utility Server, you will be running in an interactive shell on a login node. The login nodes provide login access for the Utility Server and support such activities as compiling, editing, and general interactive use by all users. The preferred method to run resource intensive executions is to use an interactive batch session.

6.4 Batch Request Submission

PBSPro ™ batch jobs are submitted via the qsub command. The format of this command is:

```
% qsub [options] <batch_script_file>
```

qsub options may be specified on the command line or imbedded in the batch script file by lines beginning with "`#PBS`".

For a more thorough discussion of PBS Batch Submission, see the PBS User Guide.

6.5 Batch Resource Directives

A complete listing of batch Resource Directives is available in the PBS User Guide.

6.6 Launch Command(s)

The following may be used to submit a job to the Utility Server.

6.6.1 Large Memory Nodes

To request a large memory node, use a variation of the following launch command:

```
% qsub -I -A $ACCOUNT -lselect=1:ncpus=32:mem=200GB
<script name>
```

The above command requests 32 CPUs and 200 GB of memory. Users may request from 128 GB to 256 GB of memory.

6.6.2 Graphics Nodes

To request a graphics node, a variation of the following launch command can be used:

```
% qsub -I -A $ACCOUNT -lselect=1:ngpus=16 <script name>
```

The above command requests 1 GPU.

6.6.3 Mixed Node Requests

To request a mixture of nodes, a variation of the following launch command can be used:

```
% qsub -I -A $ACCOUNT -lselect=1:ncpus=16:ngpus=1 <script
name>
```

The above command requests 16 CPUs and 1 GPU.

6.7 Sample Scripts

6.7.1 Large Memory Sample Script

The following script shows how to request the same resources as shown in the above section except in a full PBS script.

```
#!/bin/csh

#PBS -l walltime=h:mm:ss

#PBS -l select=ncpus=32

#PBS -l mem=200GB

#PBS -l job_type={SMP|MPI|MIX}

#PBS -j oe

#PBS -m abe

#PBS -M user1@afrl.hpc.mil, user2@your.domain

#PBS -A account_number (16-character account number or
$ACCOUNT)

#

#  Change to $WORKDIR directory and copy input file.

#  We're using the copy command to pull information from

#  $CENTER.

#

cd $WORKDIR

cp {directory in $CENTER} ../{output filename}.

#

#  Run the analysis.

date

./my_job
```

```
date

#

#  First, tar up $WORKDIR to compress

#  the data and make it easier to transfer, then we use the

#  copy command to push it to $CENTER.

#

tar cvf ../{output filename}.tar .

cp {directory in $WORKDIR} ../{output filename}.tar

#

#  Exit the script.
```

6.7.2 Graphics Node Sample Script

The following script shows how to request the same resources as shown in the above section 6.7.2 except in a full PBS script.

```
#!/bin/csh

#PBS -l walltime=h:mm:ss

#PBS -l select=ngpus=16

#PBS -l job_type={SMP|MPI|MIX}

#PBS -j oe

#PBS -m abe

#PBS -M user1@afrl.hpc.mil, user2@your.domain

#PBS -A account_number (16-character account number or
$ACCOUNT)

#

#  Change to $WORKDIR directory and copy input file.

#  We're using the copy command to pull information from
```

```
#  $CENTER.
#
cd $WORKDIR
cp {directory in $CENTER} {filename}
#
#  Run the analysis.
date
./my_job
date
#
#  First, tar up $WORKDIR to compress
#  the data and make it easier to transfer, then we use the
#  cp command to push it to $CENTER.
#
tar cvf ../{output filename}.tar .
cp {directory in $WORKDIR} ../{output filename}.tar
#
#  Exit the script.
```

6.7.3 Mixed Node Sample Script

The following script shows how to request the same resources as shown in the above section 6.6.3 except in a full PBS script.

```
#!/bin/csh
#PBS -l walltime=h:mm:ss
#PBS -l select=ncpus=32
```

```
#PBS -l select=ngpus=16

#PBS -l job_type={SMP|MPI|MIX}

#PBS -j oe

#PBS -m abe

#PBS -M user1@afrl.hpc.mil, user2@your.domain

#PBS -A account_number (16-character account number or
$ACCOUNT)

#

#  Change to $WORKDIR directory and copy input file.

#  We're using the copy command to pull information from

#  $CENTER.

#

cd $WORKDIR

cp {directory in $CENTER} ../{output filename}.tar

#

#  Run the analysis.

date

./my_job

date

#

#  First, tar up $WORKDIR to compress

#  the data and make it easier to transfer, then we use the

#  put command to push it to $CENTER. The -C option works

#  the same way here as it did above, except in this case,
```

```
#  you would give it a path to where you wanted to store

#  the files.

#

tar cvf ../{output filename}.tar .

put -C {directory in $WORKDIR} ../{output filename}.tar

#

#  Exit the script.
```

7 Software Resources

7.1 Application Software

All COTS software packages can be found in the /cots/install/ directory. The general rule for all COTS software packages is that the three latest versions will be maintained on our systems. For convenience, modules are also available for most COTS software packages.

7.2 Visualization Software

The Utility Server includes the following visualization applications:

- Ensight Suite
- Fieldview
- Matlab
- NCAR Graphics Library
- ParaView
- Tecplot
- VisIT Visualization Tool
- ezVIZ

These applications automatically take advantage of the GPUs for graphics acceleration when run within the PKIVNC secure remote desktop application.

7.3 GPGPU Computing

The Utility Server includes graphics nodes which can be used for General Purpose computing on GPUs (GPGPU). Users may write a GPGPU program using CUDA or OpenCL. The Data Analysis and Assessment Center (DAAC) provides GPGPU tutorials on the DAAC website at the following link:

7.4 Sample Code Repository

The Sample Code Repository is a directory that contains examples for COTS batch scripts, building and using serial and parallel programs, data management, and accessing and using serial and parallel math libraries. The `$SAMPLES_HOME` environment variable contains the path to this area, and is automatically defined in your login environment. Below is a listing of the examples provided in the Sample Code Repository on the Utility Server.